5

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/784,125	02/20/2004	Douglas J. Koslow	CA7037682001	5265

23639 7590 03/20/2007
BINGHAM, MCCUTCHEN LLP
2020 K STREET N.W.
18 FLOOR
WASHINGTON, DC 20006

EXAMINER	
JACOB, MARY C	
ART UNIT	PAPER NUMBER
2123	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/20/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/784,125	KOSLOW ET AL.	
	Examiner	Art Unit	
	Mary C. Jacob	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication even if timely filed may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 February 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *20 February 2004* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-30 have been presented for examination.

Drawings

2. The drawings are objected to because of the following. Figure 3 contains words that are typed over lines, making the drawing appear crowded and unclear. Figure 3 as well as Figure 5 contains text that has been "cut off" on the bottom, for example, element 303 in Figure 3 and element 1406 in Figure 5.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

3. The disclosure is objected to because of the following informalities. Appropriate correction is required.

4. The disclosure is objected to because it contains an embedded hyperlink and/or other form of browser-executable code. See page 2, line 2. Applicant is required to delete the embedded hyperlink and/or other form of browser-executable code. See MPEP § 608.01.

5. Paragraph 0032, line 4 refers to element 304 in Figure 4, however, it appears it should refer to element 404.

Claim Objections

6. Claims 1, 21-23, 27 and 28 are objected to because of the following informalities. Appropriate correction is required.

7. Claim 1, line 3, recites "a simulator handling the HDL portion". It is requested that the word "handling" be revised to further clarify what the simulator is doing. For example, the simulator "simulating" or "executing" the HDL portion.

8. Claim 1, line 4, "general language portion" would be better if written, "general programming language portion".

9. Claims 21, 22 and 23 recite, "The method of claim 18 the act...", it would be better if written, "The method of claim 18 in which the act..."

10. Claim 21, lines 2-3 recites, "a message sent to the processing of the second language portion". It is unclear what "the processing" is referring to and should be revised for clarity.

11. Claim 27, line 5, recites "a simulator handling the HDL portion". It is requested that the word "handling" be revised to further clarify what the simulator is doing. For example, the simulator "simulating" or "executing" the HDL portion.

12. Claim 27, line 6, "general language portion" would be better if written, "general programming language portion".

13. Claim 28, line 6, recites "a simulator handling the HDL portion". It is requested that the word "handling" be revised to further clarify what the simulator is doing. For example, the simulator "simulating" or "executing" the HDL portion.

14. Claim 28, lines 4-5, "general language portion" would be better if written, "general programming language portion".

Claim Rejections - 35 USC § 112

15. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

16. Claims 1-17, 24, 27, 28 and 30 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

17. Claim 1 recites "a simulator request" and "an external debugger" in line 5. It is unclear whether the "request" from "a simulator" is from the same "simulator" in line 3,

or a different simulator. It is also unclear whether "an external debugger" refers to the same "external debugger" in line 4.

18. Claims 1, 27 and 28 are directed to a method, process and system "for simultaneous debugging of an electrical design". However, the steps that follow are directed to "interrupting a simulator", "handling a simulator request with an external debugger" and "executing the request processing function at the simulator". It is unclear how these limitations "debug" the "electrical design".

19. Claims 14 and 24 recite the limitation "call *expr*". It appears that "*expr*" is some expression that can be variable based on some expression that can be defined by a user to be called by a software function. The variability of this statement "*expr*" is therefore vague and indefinite since it does not define an actual expression that is being called.

20. Claims 27 and 30 recite, "a computer program product comprising a computer usable medium having executable code to execute a process" and "a computer program product comprising a computer usable medium having executable code to execute a method". Although it is recited that the medium has code that will execute, there is no recitation of a computer actually causing the code to execute, thereby implementing the process or method steps that are further recited.

21. Claim 27 recites "a simulator request" and "an external debugger" in line 7. It is unclear whether the "request" from "a simulator" is from the same "simulator" in line 5, or a different simulator. It is also unclear whether "an external debugger" refers to the same "external debugger" in line 6.

22. Claim 28 recites "a simulator request" and "an external debugger" in line 6. It is unclear whether the "request" from "a simulator" is from the same "simulator" in line 3, or a different simulator. It is also unclear whether "an external debugger" refers to the same "external debugger" in line 4.

23. Claim 30 recites, "A system for simultaneous processing of a design..." and further recites, "the method comprising" in line 4. It is unclear what is being claimed, either "a system" or "the method".

Claim Rejections - 35 USC § 101

24. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

25. Claims 1-30 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

26. Claims 1-17 and 28 are directed to a method and system for simultaneous debugging of an electrical design. Claims 18-26 are directed to a method for simultaneous processing of a design that is based upon multiple programming languages. This claimed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring phenomenon) since it fails to produce a useful, concrete and tangible result. Specifically, the claimed subject matter does not produce a tangible result because the claimed subject matter fails to produce a result that is limited to having a real world value rather than a result that may be

interpreted to be abstract in nature, as, for example, a thought, a computation or manipulated data. More specifically, the claimed subject matter provides for "executing the request processing function at the simulator to respond to the simulator request" and "executing the request processing function at the first language portion to process the one or more waiting requests". These produced "results" remain in the abstract and thus, fails to achieve the required status of having a real world value. The "executing" of the requests is a result that remains embodied within the computer and does not appear to produce an output that can be used by a user or that is applied in a practical application.

27. Claims 27 and 29 are is directed to "a computer program product comprising a computer usable medium having executable code to execute a process", "the process comprising..." and "a computer program product comprising a computer usable medium having executable code to execute a method", "the method comprising..." Paragraph 0041 of the specification recites that the term "computer useable medium" can take the form of "transmission media" that can take the form of "carrier waves" and "light waves", which is directed to a claim to an improper computer-readable medium. A claim reciting a signal encoded with functional descriptive material fails to fall within any of the categories of patentable subject matter west forth under 35 USC 101. Further, the claimed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring phenomenon) since it fails to produce a useful, concrete and tangible result. Specifically, the claimed subject matter does not produce a tangible result because the claimed subject matter fails to produce a result that is limited

to having a real world value rather than a result that may be interpreted to be abstract in nature, as, for example, a thought, a computation or manipulated data. More specifically, the claimed subject matter provides for "executing the request processing function at the simulator to respond to the simulator request" and "executing the request processing function at the first language portion to process the one or more waiting requests". These produced "results" remain in the abstract and thus, fails to achieve the required status of having a real world value. The "executing" of the requests is a result that remains embodied within the computer and does not appear to produce an output that can be used by a user or that is applied in a practical application.

28. Claim 30 recites "a system" and "the method comprising". It is unclear whether "a system" or "the method" is being claimed, therefore, fails to fall within any of the categories of patentable subject matter west forth under 35 USC 101. Further, the claimed subject matter lacks a practical application of a judicial exception (law of nature, abstract idea, naturally occurring phenomenon) since it fails to produce a useful, concrete and tangible result. Specifically, the claimed subject matter does not produce a tangible result because the claimed subject matter fails to produce a result that is limited to having a real world value rather than a result that may be interpreted to be abstract in nature, as, for example, a thought, a computation or manipulated data. More specifically, the claimed subject matter provides for "executing the request processing function at the first language portion to process the one or more waiting requests". This produced "result" remains in the abstract and thus, fails to achieve the required status of having a real world value. The "executing" of the requests is a result that remains

embodied within the computer and does not appear to produce an output that can be used by a user or that is applied in a practical application.

Claim Rejections - 35 USC § 103

29. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

30. Claims 1-7, 12-17, 27 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hollander in view of Martinolle et al ("A Procedural Language Interface for VHDL", Proceedings of the Verilog HDL Conference and VHDL International Users Forum, 1998, pages 32-38).

31. As to Claims 1, 27 and 28, Hollander teaches: a method for simultaneous debugging (column 10, lines 39-43, lines 50-58; column 11, lines 2-5) of an electrical

design having both an HDL portion (Figure 5, element 170; column 10, lines 35-36) and a general programming language portion (Figure 5, elements 172, 163; column 10, lines 24-28), comprising: interrupting a simulator handling the HDL portion, the simulator interrupted by an external debugger (column 5, lines 44-48; column 10, lines 12-13, lines 44-50), the external debugger for debugging the general language portion (column 10, lines 50-58; column 11, lines 2-16) and handling feedback from the simulator with the external debugger (column 7, lines 18-21; column 8, lines 18-19) and requesting information from the simulator (Figure 4; column 9, lines 58-65).

32. Hollander does not expressly teach: handling a simulator request with an external debugger, the external debugger calling a request processing function at the simulator, and executing the request processing function at the simulator to respond to the simulator request.

33. Martinolle et al teaches that the VHDL community recognized that a standard C procedural interface was needed to complement VHDL, and therefore teaches a VHDL interface, VHPI, that is very similar to the Verilog Procedural Interface (VPI), that is essential to use in a mixed language design (conclusion, lines 1-9). The VHPI as taught by Martinolle et al has interaction capabilities with a simulation wherein a C model or application using VHPI can communicate with a simulator, the interactions including handling a simulator request with an external debugger (section 2.4, lines 3-5, section 3, paragraph 1, lines 7-11), the external debugger calling a request processing function at the simulator (section 2.4, lines 15-40), and executing the request processing function at the simulator to respond to the simulator request (section 2.4, lines 15-40).

34. Hollander and Martinolle et al are analogous art since they are both directed to interfacing between a simulator having a HDL portion and a general language portion.

35. It would have been obvious to one of ordinary skill at the time the invention was made to modify the method for simultaneous debugging of an electrical design having both an HDL portion and a general programming language portion as taught in Hollander to further include the handling a simulator request with an external debugger (section 2.4, lines 3-5, section 3, paragraph 1, lines 7-11), the external debugger calling a request processing function at the simulator (section 2.4, lines 15-40), and executing the request processing function at the simulator to respond to the simulator request (section 2.4, lines 15-40) as taught in Martinolle et al since Martinolle et al teaches that the VHDL community recognized that a standard C procedural interface was needed to complement VHDL, and therefore teaches a VHDL interface, VHPI, that is very similar to the Verilog Procedural Interface (VPI), that is essential to use in a mixed language design (conclusion, lines 1-9).

36. As to Claim 2, Hollander in view of Martinolle et al teach: the simulator request accesses a portion of the HDL portion (Hollander: column 7, lines 27-28; column 13, lines 38-41).

37. As to Claim 3, Hollander in view of Martinolle et al teach: the simulator request accesses HDL signal values (Hollander: column 9, lines 58-65; Martinolle et al: section 2.4, lines 37-40; section 3, paragraph 2, lines 7-9).

38. As to Claim 4, Hollander in view of Martinolle et al teach: the simulator request accesses HDL design hierarchy (Martinolle et al: Introduction, paragraph 2, lines 1-3; section 3, paragraph 1, lines 3-7, paragraph 2, lines 1-7).

39. As to Claim 5, Hollander in view of Martinolle et al teach: the simulator request operates simulator functionality (Martinolle et al: section 2.4, lines 2-7).

40. As to Claim 6, Hollander in view of Martinolle et al teach: the general programming language portion comprises C, C++, or SystemC code (Hollander: column 10, lines 27-28; Martinolle et al: section 2.4, lines 3-5).

41. As to Claim 7, Hollander in view of Martinolle et al teach: the HDL portion comprises VHDL or Verilog (Hollander: column 6, lines 37-39, lines 61-63; column 10, lines 34-35; Martinolle et al: section 2.4, lines 2-5; Introduction, paragraph 3, lines 1-4).

42. As to Claim 12, Hollander in view of Martinolle et al teach: the simulator request is generated at a simulator GUI (Hollander: column 10, line 59-column 11, line 8, Figure 6; Martinolle et al teaches: section 2.4, lines 30-40).

43. As to Claim 13, Hollander in view of Martinolle et al teach: the response to the simulator request is displayed at the simulator GUI (Hollander: column 10, line 59-column 11, line 8, Figure 6).

44. As to Claim 14, Hollander in view of Martinolle et al teach: the external debugger calls the request processing function at the simulator with the following statement: call expr (Martinolle et al: section 2.4, lines 15-40).

45. As to Claims 15, Hollander in view of Martinolle et al teach: the simulator request is routed through a debugger GUI for the external debugger (Hollander: column 10, line 59-column 11, line 8, Figure 6; Martinolle et al teaches: section 2.4, lines 30-40).

46. As to Claim 16, Hollander in view of Martinolle et al teach: the simulator request is directly routed to the external debugger (Hollander: column 10, line 59-column 11, line 8, Figure 6; Martinolle et al teaches: section 2.4, lines 30-40).

47. As to Claim 17, Hollander in view of Martinolle et al teach: the request processing function is set up ahead of time at the simulator to handle anticipated simulator requests (Hollander: column 5, lines 44-48; column 11, lines 6-8; Martinolle et al: section 2.4, lines 5-7, 15-17).

48. Claims 8-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hollander in view of Martinolle et al as applied to claim 1 above, and further in view of Chan (US Patent 6,466, 898).

49. Hollander in view of Martinolle et al teach a method for simultaneous debugging of an electrical design wherein an external debugger calls a request processing function at a simulator (Martinolle et al: section 2.4, lines 15-40).

50. Hollander in view of Martinolle et al do not expressly teach (claim 8) the action of having the external debugger call the request processing function is based upon recognition of a waiting simulator request, (claim 9) recognition of the waiting simulator request is based upon a message sent to the external debugger, (claim 10) recognition of the waiting simulator request is based upon a periodic check of a simulator request

wait queue, (claim 11) recognition of the waiting simulator request is based on whether a threshold number of simulator requests are waiting in a simulator request wait queue.

51. Chan teaches a novel concurrent, multi-threaded algorithm to accelerate the execution of logic simulation of HGL designs that supports both VHDL and Verilog HDL design languages in a single platform (column 4, lines 5-16), wherein the execution of logic simulation events includes wherein (claim 8) the call of a request processing function is based upon recognition of a waiting simulator request (column 7, lines 1-7, lines 32-39; Figure 3, element 14), (claim 9) wherein the recognition of the waiting simulator request is based upon a message sent to the logic simulation program (column 7, lines 1-7, lines 32-39; Figure 3, element 14), wherein (claim 10) recognition of the waiting simulator request is based upon a periodic check of a simulator request wait queue (column 7, lines 1-7, lines 32-46; Figure 3, element 14), and wherein (claim 11) recognition of the waiting simulator request is based on whether a threshold number of simulator requests are waiting in a simulator request wait queue (column 20, lines 55-66; Figure 17, element 89).

52. Hollander in view of Martinolle et al and Chan are analogous art since they are both directed to the simulation of a HDL design.

53. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the method for simultaneous debugging of an electrical design wherein an external debugger calls a request processing function at a simulator as taught by Hollander in view of Martinolle et al to further include the call of a request processing function is based upon recognition of a waiting simulator request, wherein

the recognition of the waiting simulator request is based upon a message sent to the logic simulation program, wherein recognition of the waiting simulator request is based upon a periodic check of a simulator request wait queue, and wherein recognition of the waiting simulator request is based on whether a threshold number of simulator requests are waiting in a simulator request wait queue as taught in Chan since Chan teaches a novel concurrent, multi-threaded algorithm to accelerate the execution of logic simulation of HGL designs that supports both VHDL and Verilog HDL design languages in a single platform (column 4, lines 5-16).

54. Claims 18-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hollander in view of Martinolle et al and further in view of Chan.

55. As to Claims 18, 29 and 30, Hollander teaches: a method for simultaneous processing of a design (column 10, lines 39-43, lines 50-58; column 11, lines 2-5) that is based upon multiple programming languages, the multiple programming languages comprising a first language portion (Figure 5, element 170; column 10, lines 35-36) and a second language portion (Figure 5, elements 172, 163; column 10, lines 24-28), in which processing of the second language portion interrupts processing of the first language portion (column 5, lines 44-48; column 10, lines 12-13, lines 44-50), the method comprising: processing the second language portion of the design causing an interruption of processing for the first language portion (column 5, lines 44-48; column 10, lines 12-13, lines 44-50);

56. Hollander does not expressly teach: determining whether there are one or more waiting requests for processing of the first language portion; handling the one or more waiting requests for processing of the first language portion by having processing of the second language portion call a request processing function at the first language portion; and executing the request processing function at the first language portion to process the one or more waiting requests.

57. Martinolle et al teaches that the VHDL community recognized that a standard C procedural interface was needed to complement VHDL, and therefore teaches a VHDL interface, VHPI, that is very similar to the Verilog Procedural Interface (VPI), that is essential to use in a mixed language design (conclusion, lines 1-9). The VHPI as taught by Martinolle et al has interaction capabilities with a simulation wherein a C model or application using VHPI can communicate with a simulator, the interactions including handling one or more requests for processing of the first language portion by having processing of the second language portion call a request processing function at the first language portion and executing the request processing function at the first language portion to process the one or more requests (section 2.4, lines 3-5, lines 15-40).

58. Hollander and Martinolle et al are analogous art since they are both directed to the logic simulation of mixed language designs.

59. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the method of simultaneous processing of a design that is based on multiple programming languages as taught in Hollander to further include the handling one or more requests for processing of the first language portion by having

processing of the second language portion call a request processing function at the first language portion and executing the request processing function at the first language portion to process the one or more requests as taught by Martinolle et al since Martinolle et al teaches that the VHDL community recognized that a standard C procedural interface was needed to complement VHDL, and therefore teaches a VHDL interface, VHPI, that is very similar to the Verilog Procedural Interface (VPI), that is essential to use in a mixed language design (conclusion, lines 1-9).

60. Hollander in view of Martinolle et al do not expressly teach: determining whether there are one or more waiting requests for processing of the first language portion.

61. Chan teaches a novel concurrent, multi-threaded algorithm to accelerate the execution of logic simulation of HGL designs that supports both VHDL and Verilog HDL design languages in a single platform (column 4, lines 5-16), wherein event-driven logic simulation, as known in the art and by the invention as taught in Chan, includes determining whether there are one or more waiting requests for processing of a portion of the mixed language design (column 7, lines 1-7, lines 32-39; Figure 3, element 14; Figure 8, element 42).

62. Hollander in view of Martinolle et al and Chan are analogous art since they are all directed to the simulation of a mixed language design.

63. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the simultaneous processing of a based on multiple programming languages as taught by Hollander in view of Martinolle et al to further include determining whether there are one or more requests waiting for processing a

portion of the design as taught in Chan since Chan teaches a novel concurrent, multi-threaded algorithm to accelerate the execution of logic simulation of HGL designs that supports both VHDL and Verilog HDL design languages in a single platform (column 4, lines 5-16) and further teaches that event driven simulation, as known in the art, includes determining whether there are one or more waiting requests for processing of a portion of the mixed language design (column 7, lines 1-7, lines 32-39; Figure 3, element 14; Figure 8, element 42).

64. As to Claim 19, Hollander in view of Martinolle et al in further view of Chan teach: the one or more waiting requests are for accessing data from the first language portion of the design (Hollander: column 5, lines 44-49; column 9, lines 58-65; Martinolle et al: section 2.4, lines 37-40; section 3, paragraph 2, lines 7-9).

65. As to Claim 20, Hollander in view of Martinolle et al in further view of Chan teach: the one or more waiting requests are for debugging the first language portion (Hollander: column 10, lines column 4, lines 45-47; column 5, lines 39-50; column 9, lines 58-65; column 10, lines 43-61).

66. As to Claim 21, Hollander in view of Martinolle et al in further view of Chan teach: the act of determining whether there are one or more waiting requests for processing of the first language portion is based upon a message sent to the processing of the second language portion (Martinolle: section 2.4, lines 2-5, lines 15-17; Chan: column 7, lines 1-7, lines 32-39; Figure 3, element 14).

67. As to Claim 22, Hollander in view of Martinolle et al in further view of Chan teach: the act of determining whether there are one or more waiting requests for processing of

the first language portion is based a periodic check of a request wait queue for the first language portion (Chan: column 7, lines 1-7, lines 32-46; Figure 3, element 14).

68. As to Claim 23, Hollander in view of Martinolle et al in further view of Chan teach: the act of determining whether there are one or more waiting requests for processing of the first language portion is based on whether a threshold number of simulator requests are waiting in a request wait queue (Chan: column 20, lines 55-66; Figure 17, element 89).

69. As to Claim 24, Hollander in view of Martinolle et al in further view of Chan teach: the request processing function is called with the following statement: call expr (Martinolle et al: section 2.4, lines 15-40).

70. As to Claim 25, Hollander in view of Martinolle et al in further view of Chan teach: processing the second language portion comprises debugging the second language portion (Hollander: column 10, lines 24-28, lines 39-42, lines 54-56; column 10, line 67- column 11, line 5; Chan: column 10, lines 37-42).

71. As to Claim 26, Hollander in view of Martinolle et al in further view of Chan teach: the request processing function is set up ahead of time to handle anticipated requests (Hollander: column 5, lines 44-48; column 11, lines 6-8; Martinolle et al: section 2.4, lines 5-7, 15-17).

Conclusion

72. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

73. Zammit et al (US Patent 7,146,300) teaches a method for co-simulating a digital circuit using a simulation engine which communicates with one or more first programming languages by means of a foreign language interface and which communicates directly with one or more second programming language.

74. Powell (US Patent 7,058,562) teaches an apparatus that performs event processing in a mixed-language simulator.

75. Flake et al (US Patent 7,035,781) teaches an HDL simulator having an automated interface to compiled or interpreted application code written in a general purpose language.

76. Schubert et al (US Patent 6,581,191) teaches techniques and systems for analysis, diagnosis and debugging fabricated hardware designs at a HDL level.

77. Yu et al (US Patent 6,263,303) teaches a simulator for simulating hardware/software behavior of embedded systems and teaches simulator event queues.

78. Bahra et al (US Patent 6,226,780) teaches a method and apparatus that can support multiple HDLs wherein a graphical design tool supports a first HDL and design data is provided in a second HDL. The invention generates an interface description for the design data written in the first HDL, and then generates a graphical design unit based on the interface description.

79. McKinney, Michael ("Integrating Perl, Tcl and C++ into Simulation-Based ASIC Verification Environments", Proceedings of the 6th IEEE International High-Level Design Validation and Test Workshop, 2001, pages 19-24) teaches a simulation based ASIC verification environment that includes Perl, Tcl and C++ wherein the Tcl gives users access to internal simulator events and variables.

80. Bombana et al ("SystemC-VHDL Co-Simulation and Synthesis in the HW Domain", Design, Automation and Test in Europe Conference and Exhibition, 2003, pages 101-105) teaches the analysis and formulation of a design methodology able to mix abstraction levels and languages for HW development, combining behavioral synthesis and SystemC/VHDL cosimulation.

81. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary C. Jacob whose telephone number is 571-272-6249. The examiner can normally be reached on M-F 7AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached on 571-272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

***

Mary C. Jacob
Examiner
AU2123

MCJ
3/15/07

ZOILA CABRERA
PRIMARY EXAMINER
TECHNOLOGY CENTER 2100

3/15/07